



Wrox Programmer to Programmer™



Beginning

# C# and .NET

2021 Edition

Benjamin Perkins and Jon D. Reid



# BEGINNING C# AND .NET

---

INTRODUCTION .....	xxvii
▶ PART I THE C# LANGUAGE	
CHAPTER 1	Introducing C# ..... 3
CHAPTER 2	Writing a C# Program..... 13
CHAPTER 3	Variables and Expressions ..... 29
CHAPTER 4	Flow Control..... 55
CHAPTER 5	More about Variables..... 79
CHAPTER 6	Functions ..... 113
CHAPTER 7	Debugging and Error Handling ..... 145
CHAPTER 8	Introduction to Object-Oriented Programming ..... 175
CHAPTER 9	Defining Classes..... 199
CHAPTER 10	Defining Class Members ..... 231
CHAPTER 11	Collections, Comparisons, and Conversions ..... 265
CHAPTER 12	Generics ..... 315
CHAPTER 13	Additional C# Techniques ..... 355
▶ PART II DATA ACCESS	
CHAPTER 14	Files..... 423
CHAPTER 15	XML and JSON..... 453
CHAPTER 16	LINQ..... 479
CHAPTER 17	Databases..... 513
▶ PART III ADDITIONAL TECHNIQUES	
CHAPTER 18	.NET and ASP.NET ..... 545
CHAPTER 19	Basic Cloud Programming ..... 591

*Continues*

<b>CHAPTER 20</b>	<b>Basic Web API and WCF Programming . . . . .</b>	<b>617</b>
<b>CHAPTER 21</b>	<b>Basic Desktop Programming . . . . .</b>	<b>663</b>
<b>APPENDIX</b>	<b>Exercise Solutions. . . . .</b>	<b>769</b>
<b>INDEX . . . . .</b>		<b>809</b>

BEGINNING  
**C# and .NET**



BEGINNING  
**C# and .NET**  
2021 EDITION

Benjamin Perkins

Jon D. Reid



Copyright © 2021 by John Wiley & Sons, Inc. All rights reserved.

Published by John Wiley & Sons, Inc., Hoboken, New Jersey.

Published simultaneously in Canada.

ISBN: 978-1-119-79578-0

ISBN: 978-1-119-79582-7 (ebk)

ISBN: 978-1-119-79583-4 (ebk)

No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning, or otherwise, except as permitted under Section 107 or 108 of the 1976 United States Copyright Act, without either the prior written permission of the Publisher, or authorization through payment of the appropriate per-copy fee to the Copyright Clearance Center, Inc., 222 Rosewood Drive, Danvers, MA 01923, (978) 750-8400, fax (978) 750-4470, or on the web at [www.copyright.com](http://www.copyright.com). Requests to the Publisher for permission should be addressed to the Permissions Department, John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030, (201) 748-6011, fax (201) 748-6008, or online at <http://www.wiley.com/go/permission>.

**Limit of Liability/Disclaimer of Warranty:** While the publisher and author have used their best efforts in preparing this book, they make no representations or warranties with respect to the accuracy or completeness of the contents of this book and specifically disclaim any implied warranties of merchantability or fitness for a particular purpose. No warranty may be created or extended by sales representatives or written sales materials. The advice and strategies contained herein may not be suitable for your situation. You should consult with a professional where appropriate. Neither the publisher nor author shall be liable for any loss of profit or any other commercial damages, including but not limited to special, incidental, consequential, or other damages.

For general information on our other products and services or for technical support, please contact our Customer Care Department within the United States at (800) 762-2974, outside the United States at (317) 572-3993 or fax (317) 572-4002.

Wiley also publishes its books in a variety of electronic formats. Some content that appears in print may not be available in electronic formats. For more information about Wiley products, visit our web site at [www.wiley.com](http://www.wiley.com).

**Library of Congress Control Number:** 2021938274

**Trademarks:** WILEY and the Wiley logo are trademarks or registered trademarks of John Wiley & Sons, Inc. and/or its affiliates, in the United States and other countries, and may not be used without written permission. All other trademarks are the property of their respective owners. John Wiley & Sons, Inc. is not associated with any product or vendor mentioned in this book.

Cover Image: ©Ben Clift Williams/EyeEm/Getty Images

Cover Design: Wiley



# ABOUT THE AUTHORS

**Benjamin Perkins** is currently employed at Microsoft in Munich, Germany, as a Senior Escalation Engineer for IIS, ASP.NET, and Azure App Services. He has been working professionally in the IT industry for over two decades. He started computer programming with QBasic at the age of 11 on an Atari 1200XL desktop computer. He takes pleasure in the challenges that troubleshooting technical issues have to offer and savors in the rewards of a well-written program. After completing high school, he joined the United States Army. After successfully completing his military service, he attended Texas A&M University in College Station, Texas, where he received a Bachelor of Business Administration in Management Information Systems. He also received a Master of Business Administration from the European University.

His roles in the IT industry have spanned the entire spectrum including programmer, system architect, technical support engineer, team leader, and mid-level management. While employed at Hewlett-Packard, he received numerous awards, degrees, and certifications. He has a passion for technology and customer service and looks forward to troubleshooting and writing more world-class technical solutions. “My approach is to write code with support in mind, and to write it once correctly and completely so we do not have to come back to it again, except to enhance it.”

Benjamin has written numerous magazine articles and training courses and is an active blogger. His catalog of books covers C# Programming, IIS, NHibernate, Open Source, and Microsoft Azure.

- Connect with Benjamin on LinkedIn: [www.linkedin.com/in/csharpguitar](http://www.linkedin.com/in/csharpguitar)
- Follow Benjamin on Twitter @csharpguitar: [twitter.com/csharpguitar](https://twitter.com/csharpguitar)
- Read Benjamin’s blog: [www.thebestcsharpprogrammerintheworld.com](http://www.thebestcsharpprogrammerintheworld.com)
- Visit Benjamin on GitHub: [github.com/benperk](https://github.com/benperk)

Benjamin is married to Andrea and has two wonderful children, Lea and Noa.

**Jon D. Reid** is a Program Manager in Research and Development for IFS AB ([www.ifs.com](http://www.ifs.com)) focusing on Field Service Management. He has coauthored many books on Microsoft technologies, including *Beginning C# 7 Programming with Visual Studio 2017*, *Fast Track C#*, *Pro Visual Studio .NET*, and many others.



# ABOUT THE TECHNICAL EDITOR

**Rod Stephens** is a long-time developer and author who has written more than 250 magazine articles and 35 books that have been translated into different languages around the world. During his career, Rod has worked on an eclectic assortment of applications in such fields as telephone switching, billing, repair dispatching, tax processing, wastewater treatment, concert ticket sales, cartography, and training for professional football teams.

Rod's popular C# Helper website ([www.csharpHelper.com](http://www.csharpHelper.com)) receives millions of hits per year and contains tips, tricks, and example programs for C# programmers. His VB Helper website ([www.vb-helper.com](http://www.vb-helper.com)) contains similar material for Visual Basic programmers.

You can contact Rod at: [RodStephens@csharpHelper.com](mailto:RodStephens@csharpHelper.com) or [RodStephens@vb-helper.com](mailto:RodStephens@vb-helper.com).



# ACKNOWLEDGMENTS

It takes a lot of work to get content into a presentable format for students and IT professionals to read and get value from. The authors indeed have technical knowledge and experiences to share, but without the technical writers, technical reviewers, developers, editors, publishers, graphic designers, the list goes on, providing their valuable input, a book of high quality could not be written. The rate of change occurs too quickly for an individual to perform all these tasks and still publish a book that is valid before the technology becomes stale. This is why authors worked together with great teams to get all the components of the book together quickly. It was done to ensure that the most up-to-date information gets to the reader while the features are still fresh and current. I would like to thank Rod Stephens for his technical review and suggestions throughout the process. Lastly, I would like to thank all the numerous people behind the scenes who helped get this book together.



# CONTENTS

INTRODUCTION

xxvii

## PART I: THE C# LANGUAGE

### CHAPTER 1: INTRODUCING C# 3

---

What Is .NET?	3
.NET Framework, .NET Standard, and .NET Core	4
Writing Programs Using .NET	5
CIL and JIT	5
Assemblies	6
Managed Code	7
Garbage Collection	7
Fitting It Together	7
Linking	8
What Is C#?	8
Applications You Can Write with C#	9
C# in This Book	10
Visual Studio	10
Visual Studio Products	10
Solutions	11

### CHAPTER 2: WRITING A C# PROGRAM 13

---

The Visual Studio Development Environment	14
Console Applications	18
The Solution Explorer	21
The Properties Window	22
The Error List Window	23
Desktop Applications	23

### CHAPTER 3: VARIABLES AND EXPRESSIONS 29

---

Basic C# Syntax	30
Basic C# Console Application Structure	33
Variables	34
Simple Types	34
Variable Naming	39

Literal Values	39
Binary Literals and Digit Separators	40
String Literals	41
<b>Expressions</b>	<b>42</b>
Mathematical Operators	43
Assignment Operators	48
Operator Precedence	49
Namespaces	50
<b>CHAPTER 4: FLOW CONTROL</b>	<b>55</b>
<b>Boolean Logic</b>	<b>56</b>
Boolean Bitwise and Assignment Operators	58
Operator Precedence Updated	60
<b>Branching</b>	<b>60</b>
The Ternary Operator	61
The if Statement	61
Checking More Conditions Using if Statements	64
The switch Statement	65
<b>Looping</b>	<b>68</b>
do Loops	68
while Loops	71
for Loops	73
Interrupting Loops	74
Infinite Loops	75
<b>CHAPTER 5: MORE ABOUT VARIABLES</b>	<b>79</b>
<b>Type Conversion</b>	<b>80</b>
Implicit Conversions	80
Explicit Conversions	82
Explicit Conversions Using the Convert Commands	84
<b>Complex Variable Types</b>	<b>87</b>
Enumerations	87
Defining Enumerations	88
Structs	91
Defining Structs	92
Arrays	94
Declaring Arrays	95
foreach Loops	98



---

Pattern Matching with switch case Expression	98
Multidimensional Arrays	102
Arrays of Arrays	104
String Manipulation	105
<b>CHAPTER 6: FUNCTIONS</b>	<b>113</b>
<hr/>	
Defining and Using Functions	114
Return Values	117
Parameters	118
Parameter Matching	121
Parameter Arrays	121
Reference and Value Parameters	123
Out Parameters	125
Tuples	126
Variable Scope	128
Variable Scope in Other Structures	131
Parameters and Return Values versus Global Data	132
Local Functions	134
The Main() Function	135
Struct Functions	137
Overloading Functions	138
Using Delegates	140
<b>CHAPTER 7: DEBUGGING AND ERROR HANDLING</b>	<b>145</b>
<hr/>	
Debugging in Visual Studio	146
Debugging in Nonbreak (Normal) Mode	147
Outputting Debugging Information	148
Tracepoints	152
Diagnostics Output versus Tracepoints	154
Debugging in Break Mode	155
Entering Break Mode	155
Monitoring Variable Content	158
Stepping through Code	161
Immediate and Command Windows	162
The Call Stack Window	163
Error Handling	163
try. . .catch. . .finally	164
Throw Expressions	172
Listing and Configuring Exceptions	172

<b>CHAPTER 8: INTRODUCTION TO OBJECT-ORIENTED PROGRAMMING</b>	<b>175</b>
<b>What Is Object-Oriented Programming?</b>	<b>176</b>
What Is an Object?	177
Properties and Fields	178
Methods	179
Everything's an Object	180
The Life Cycle of an Object	180
Constructors	180
Destructors	181
Static and Instance Class Members	181
Static Constructors	181
Static Classes	182
<b>OOP Techniques</b>	<b>182</b>
Interfaces	182
Disposable Objects	184
Inheritance	184
Polymorphism	187
Interface Polymorphism	188
Relationships between Objects	189
Containment	189
Collections	190
Operator Overloading	191
Events	191
Reference Types versus Value Types	192
<b>OOP in Desktop Applications</b>	<b>192</b>
<b>CHAPTER 9: DEFINING CLASSES</b>	<b>199</b>
<b>Class Definitions in C#</b>	<b>200</b>
Interface Definitions	202
<b>System.Object</b>	<b>205</b>
<b>Constructors and Destructors</b>	<b>207</b>
Constructor Execution Sequence	209
<b>OOP Tools in Visual Studio</b>	<b>212</b>
The Class View Window	212
The Object Browser	214
Adding Classes	216
Class Diagrams	217

---

Class Library Projects	219
Interfaces versus Abstract Classes	223
Struct Types	225
Shallow Copying versus Deep Copying	227
<b>CHAPTER 10: DEFINING CLASS MEMBERS</b>	<b>231</b>
<hr/>	
<b>Member Definitions</b>	<b>232</b>
Defining Fields	232
Defining Methods	233
Defining Properties	234
Tuple Deconstruction	239
Refactoring Members	240
Automatic Properties	241
<b>Additional Class Member Topics</b>	<b>242</b>
Hiding Base Class Methods	242
Calling Overridden or Hidden Base Class Methods	244
The this Keyword	244
Using Nested Type Definitions	245
<b>Interface Implementation</b>	<b>247</b>
Implementing Interfaces in Classes	248
Explicit Interface Member Implementation	249
Additional Property Accessors	249
<b>Partial Class Definitions</b>	<b>250</b>
<b>Partial Method Definitions</b>	<b>251</b>
<b>Example Application</b>	<b>252</b>
Planning the Application	253
The Card Class	253
The Deck Class	253
Writing the Class Library	253
Adding the Suit and Rank Enumerations	254
Adding the Card Class	256
Adding the Deck Class	258
A Client Application for the Class Library	261
<b>The Call Hierarchy Window</b>	<b>262</b>
<b>CHAPTER 11: COLLECTIONS, COMPARISONS, AND CONVERSIONS</b>	<b>265</b>
<hr/>	
<b>Collections</b>	<b>266</b>
Using Collections	267
Defining Collections	272

---

Indexers	273
Adding a CardCollection to CardLib	275
Keyed Collections and IDictionary	278
Iterators	279
Iterators and Collections	284
Deep Copying	285
Adding Deep Copying to CardLib	286
<b>Comparisons</b>	<b>288</b>
Type Comparisons	288
Boxing and Unboxing	289
The is Operator	290
Pattern Matching with the is Operator Pattern Expression	293
Value Comparisons	294
Operator Overloading	294
Adding Operator Overloads to CardLib	299
The IComparable and IComparer Interfaces	304
Sorting Collections	306
<b>Conversions</b>	<b>309</b>
Overloading Conversion Operators	310
The as Operator	311
<b>CHAPTER 12: GENERICS</b>	<b>315</b>
<b>What Are Generics?</b>	<b>316</b>
<b>Using Generics</b>	<b>317</b>
Nullable Types	317
Operators and Nullable Types	318
The ?? Operator	319
The ?. Operator	320
Working with Nullable Types	321
The System.Collections.Generic Namespace	325
List<T>	326
Sorting and Searching Generic Lists	327
Dictionary<K, V>	333
Modifying CardLib to Use a Generic Collection Class	334
<b>Defining Generic Types</b>	<b>335</b>
Defining Generic Classes	336
The default Keyword	338
Constraining Types	338
Inheriting from Generic Classes	344
Generic Operators	345

---

Generic Structs	346
Defining Generic Interfaces	346
Defining Generic Methods	346
Defining Generic Delegates	348
<b>Variance</b>	<b>348</b>
Covariance	349
Contravariance	350
<b>CHAPTER 13: ADDITIONAL C# TECHNIQUES</b>	<b>355</b>
<hr/>	
<b>The :: Operator and the Global Namespace Qualifier</b>	<b>356</b>
<b>Custom Exceptions</b>	<b>357</b>
Adding Custom Exceptions to CardLib	358
<b>Events</b>	<b>359</b>
What Is an Event?	359
Handling Events	361
Defining Events	363
Multipurpose Event Handlers	367
The EventHandler and Generic EventHandler<T> Types	370
Return Values and Event Handlers	370
Anonymous Methods	370
<b>Expanding and Using CardLib</b>	<b>371</b>
<b>Attributes</b>	<b>380</b>
Reading Attributes	380
Creating Attributes	381
<b>Initializers</b>	<b>382</b>
Object Initializers	383
Collection Initializers	385
<b>Type Inference</b>	<b>388</b>
<b>Anonymous Types</b>	<b>390</b>
<b>Dynamic Lookup</b>	<b>394</b>
The dynamic Type	395
<b>Advanced Method Parameters</b>	<b>399</b>
Optional Parameters	399
Optional Parameter Values	400
The OptionalAttribute Attribute	400
Optional Parameter Order	401
Named Parameters	401
<b>Lambda Expressions</b>	<b>405</b>
Anonymous Methods Recap	405
Lambda Expressions for Anonymous Methods	407

Lambda Expression Parameters	410
Lambda Expression Statement Bodies	411
Lambda Expressions as Delegates and Expression Trees	412
Lambda Expressions and Collections	413

## PART II: DATA ACCESS

<b>CHAPTER 14: FILES</b>	<b>423</b>
File Classes for Input and Output	424
The File and Directory Classes	424
The FileInfo Class	426
The DirectoryInfo Class	428
Path Names and Relative Paths	428
Streams	429
Classes for Using Streams	429
The FileStream Object	429
File Position	430
Reading Data	432
Writing Data	434
The StreamWriter Object	436
The StreamReader Object	439
Reading Data	440
Asynchronous File Access	441
Reading and Writing Compressed Files	442
Monitoring the File System	445
<b>CHAPTER 15: XML AND JSON</b>	<b>453</b>
XML Basics	454
JSON Basics	455
XML Schemas	455
XML Document Object Model	458
The XmlDocument Class	458
The XmlElement Class	459
Changing the Values of Nodes	463
Inserting New Nodes	465
Deleting Nodes	467
Selecting Nodes	469
Searching XML with XPath	469
JSON Serialization and Deserialization	473

---

<b>CHAPTER 16: LINQ</b>	<b>479</b>
<b>LINQ to XML</b>	<b>480</b>
LINQ to XML Functional Constructors	480
Working with XML Fragments	483
<b>LINQ Providers</b>	<b>486</b>
<b>LINQ Query Syntax</b>	<b>486</b>
Declaring a Variable for Results Using the var Keyword	488
Specifying the Data Source: from Clause	489
Specify Condition: where Clause	489
Selecting Items: select Clause	490
Finishing Up: Using the foreach Loop	490
Deferred Query Execution	490
<b>LINQ Method Syntax</b>	<b>490</b>
LINQ Extension Methods	491
Query Syntax versus Method Syntax	491
Lambda Expressions	492
<b>Ordering Query Results</b>	<b>494</b>
<b>Understanding the orderby Clause</b>	<b>495</b>
<b>Querying a Large Data Set</b>	<b>496</b>
<b>Using Aggregate Operators</b>	<b>498</b>
<b>Using the Select Distinct Query</b>	<b>502</b>
<b>Ordering by Multiple Levels</b>	<b>504</b>
<b>Using Group Queries</b>	<b>506</b>
<b>Using Joins</b>	<b>508</b>
<b>CHAPTER 17: DATABASES</b>	<b>513</b>
Using Databases	514
Entity Framework	514
Code-First versus Database-First	514
Migrations and Scaffolding	515
Install SQL Server Express LocalDB	515
A Code-First Database	516
Exploring Your Database	527
Navigating Database Relationships	530
Creating and Querying XML from an Existing Database	536

**PART III: ADDITIONAL TECHNIQUES**

<b>CHAPTER 18: .NET AND ASP.NET</b>	<b>545</b>
<b>Cross-Platform Basics and Key “Must Know” Terms</b>	<b>547</b>
<b>What was .NET Standard?</b>	<b>549</b>
Shared Project, PCL, and .NET Standard	551
Building and Packaging a .NET Standard Library	553
<b>Referencing and Targeting .NET</b>	<b>557</b>
<b>What was .NET Core?</b>	<b>558</b>
Cross Platform	558
Open Source	559
Optimized for the Cloud	560
Performance	560
Modular Design	561
Self-Contained Deployment Model	562
<b>Porting from .NET Framework to .NET</b>	<b>565</b>
Identifying Third-Party Dependencies	567
Understanding Which Features Are Not Available	567
Upgrading the Current .NET Framework Target	567
<b>Overview of Web Applications</b>	<b>568</b>
<b>Which ASP.NET to Use and Why</b>	<b>569</b>
ASP.NET Web Forms	570
Server Controls	572
Input Validation	573
State Management	574
Authentication and Authorization	575
ASP.NET Web Site versus ASP.NET Web Applications	575
ASP.NET MVC/ASP.NET Core Web App MVC	577
ASP.NET Core Web API	579
ASP.NET Core Web App	580
IIS and Kestrel	581
Blazor App and Razor Pages	582
Input Validation	584
State Management	585
Authentication and Authorization	586
Dependency Injection	586
<b>CHAPTER 19: BASIC CLOUD PROGRAMMING</b>	<b>591</b>
<b>The Cloud, Cloud Computing, and the Cloud Optimized Stack</b>	<b>592</b>
<b>Cloud Patterns and Best Practices</b>	<b>595</b>



---

Using Microsoft Azure C# Libraries to Create a Storage Container	597
Creating an ASP.NET Core Web Application That Uses the Storage Container	607
<b>CHAPTER 20: BASIC WEB API AND WCF PROGRAMMING</b>	<b>617</b>
<hr/>	
Creating an ASP.NET Core Web API	617
Consuming an ASP.NET Core Web API	622
What Is REST?	629
What Is WCF?	630
WCF Concepts	631
WCF Communication Protocols	631
Addresses, Endpoints, and Bindings	632
Contracts	634
Message Patterns	634
Behaviors	635
Hosting	635
WCF Programming	635
The WCF Test Client	642
Defining WCF Service Contracts	644
Data Contracts	645
Service Contracts	645
Operation Contracts	646
Message Contracts	647
Fault Contracts	647
Self-Hosted WCF Services	652
<b>CHAPTER 21: BASIC DESKTOP PROGRAMMING</b>	<b>663</b>
<hr/>	
XAML	664
Separation of Concerns	665
XAML in Action	665
Namespaces	666
Code-Behind Files	667
The Playground	667
WPF Controls	669
Properties	670
Dependency Properties	673
Attached Properties	673
Events	674
Handling Events	675

Routed Events	676
Routed Commands	676
Control Types	679
<b>Control Layout</b>	<b>679</b>
Basic Layout Concepts	679
Stack Order	679
Alignment, Margins, Padding, and Dimensions	680
Border	680
Visual Debugging Tools	681
Layout Panels	682
Canvas	682
DockPanel	684
StackPanel	686
WrapPanel	687
Grid	688
<b>The Game Client</b>	<b>691</b>
The About Window	691
Designing the User Interface	692
The Image Control	692
The Label Control	692
The TextBlock Control	693
The Button Control	693
The Options Window	696
The TextBox Control	697
The CheckBox Control	698
The RadioButton Control	699
The ComboBox Control	700
The TabControl	701
Handling Events in the Options Window	705
Data Binding	707
The DataContext	708
Binding to Local Objects	708
Static Binding to External Objects	709
Dynamic Binding to External Objects	710
Starting a Game with the ListBox Control	712
<b>Creating and Styling Controls</b>	<b>716</b>
Styles	716
Templates	717
Triggers	719
Animations	720

---

<b>WPF User Controls</b>	<b>721</b>
Implementing Dependency Properties	722
<b>The Main Window</b>	<b>736</b>
The Menu Control	736
Routed Commands with Menus	736
<b>Putting It All Together</b>	<b>741</b>
Refactoring the Domain Model	741
The View Model	748
Completing the Game	757
<b>APPENDIX: EXERCISE SOLUTIONS</b>	<b>769</b>

---

<i>INDEX</i>	<i>809</i>
--------------	------------



# INTRODUCTION

**THE C# LANGUAGE WAS UNVEILED TO THE WORLD** when Microsoft announced the first version of its .NET Framework in 2002. Since then, its popularity has rocketed, and it has arguably become the language of choice for desktop, web, cloud, and cross-platform developers who use .NET. Part of the appeal of C# comes from its clear syntax, which derives from C/C++ but simplifies some things that have previously discouraged some programmers. Despite this simplification, C# has retained the power of C++, and there is no reason now not to move into C#. The language is not difficult, and it is a great one with which to learn elementary programming techniques. This ease of learning combined with the capabilities of the .NET Framework make C# an excellent way to start your programming career.

The latest release of C# is C# 9 (included with .NET 5.0 and .NET Framework 4.8), which builds on the existing successes and adds even more attractive features. The latest releases of both Visual Studio and Visual Studio Code line of development tools also bring many tweaks and improvements to make your life easier and to dramatically increase your productivity.

This book is intended to teach you about all aspects of C# programming, including the language itself, desktop, cloud, and cross-platform programming, making use of data sources, and some new and advanced techniques. You will also learn about the capabilities of Visual Studio and all the ways that this product can aid your application development.

The book is written in a friendly, mentor-style fashion, with each chapter building on previous ones, and every effort is made to ease you into advanced techniques painlessly. At no point will technical terms appear from nowhere to discourage you from continuing; every concept is introduced and discussed as required. Technical jargon is kept to a minimum, but where it is necessary, it, too, is properly defined and laid out in context.

The authors of this book are both experts in their field and are enthusiastic in their passion for the C# language and .NET. Nowhere will you find two people better qualified to take you under their collective wing and nurture your understanding of C# from first principles to advanced techniques. Along with the fundamental knowledge it provides, this book is packed full of helpful hints, tips, exercises, and full-fledged example code (available for download on this book's web page at [www.wiley.com](http://www.wiley.com) and at [github.com/benperk/BeginningCSharpAndDotNET](https://github.com/benperk/BeginningCSharpAndDotNET)) that you will find yourself returning to repeatedly as your career progresses.

We pass this knowledge on without begrudging it and hope that you will be able to use it to become the best programmer you can be. Good luck, and all the best!

## WHO THIS BOOK IS FOR

This book is for everyone who wants to learn how to program in C# using .NET. It is for absolute beginners who want to give programming a try by learning a clean, modern, elegant programming

language. But it is also for people familiar with other programming languages who want to explore the .NET platform, as well as for existing .NET developers who want to give Microsoft's .NET flagship language a try.

## WHAT THIS BOOK COVERS

The early chapters cover the language itself, assuming no prior programming experience. If you have programmed in other languages before, much of the material in these chapters will be familiar. Many aspects of C# syntax are shared by other languages, and many structures are common to practically all programming languages (such as looping and branching structures). However, even if you are an experienced programmer, you will benefit from looking through these chapters to learn the specifics of how these techniques apply to C#.

If you are new to programming, you should start from the beginning, where you will learn basic programming concepts and become acquainted with both C# and the .NET platform that underpins it. If you are new to .NET but know how to program, you should read Chapter 1 and then skim through the next few chapters before continuing with the application of the C# language. If you know how to program but have not encountered an object-oriented programming language before, you should read the chapters from Chapter 8 onward.

Alternatively, if you already know the C# language, you might want to concentrate on the chapters dealing with the most recent .NET and C# language developments, specifically the chapters on collections, generics, and C# language enhancements (Chapters 11 and 12).

The chapters in this book have been written with a dual purpose in mind: They can be read sequentially to provide a complete tutorial in the C# language, and they can be dipped into as required for reference material.

In addition to the core material, starting with Chapter 3 most chapters also include a selection of exercises at the end, which you can work through to ensure that you have understood the material. The exercises range from simple multiple choice or true/false questions to more complex exercises that require you to modify or build applications. The answers to all the exercises are provided in the Appendix. You can also find these exercises as part of the [wiley.com](http://wiley.com) code downloads on this book's page at [www.wiley.com](http://www.wiley.com).

Every chapter receives an overhaul with every new release of C# and .NET, the less relevant material is removed, and new material added. All the code has been tested against the latest version of the development tools used, and all the screenshots have been retaken in the most current version of the Windows OS to provide the most current windows and dialog boxes. New highlights of this edition include the following:

- Additional and improved code examples for you to try out
- Examples of programming ASP.NET Core for running cross-platform
- Examples of programming cloud applications, using Azure SDK to create and access cloud resources